

## METHOD AND SYSTEM FOR DETERMINISTIC MATCHING OF OBJECTS AND EVENTS WHICH ARE NOT UNIQUELY IDENTIFIED

### Background of the Invention

#### [0000] Definitions:

<b>S:</b>	"the set S"
<b> S :</b>	"the number of elements in S."
<b> x :</b>	"the number of characters in 'x'"
<b>S U s:</b>	"the addition of element s to S"
<b>&lt;v1, ... vn&gt;:</b>	"the n-tuple of values vi"
<b>→:</b>	"maps to"

**[0001]** The subject invention relates to deterministic matching of objects or events in a first set with objects or events in a second set when the objects or events are not all uniquely identified. More particularly it relates to matching letters to events which occur during the delivery of such letters. By "deterministic matching" herein is meant matching of events (or elements in the second set) to letters (or elements in the first set) if and only if sufficient information is available to uniquely associate events with particular letters.

**[0002]** It is often useful to be able to match objects or events in a first set with objects or events in other sets. For example, a hypothetical automobile assembly process may require that a particular engine be assembled with a particular style body, paint color, etc. to produce a car as ordered. An effective way to achieve this is to associate codes with the objects or events to be matched so that objects or events which match are associated with the same code. This a relatively simple process where the codes suffice to uniquely identify the objects or events within each set. However, if the codes are not sufficient to uniquely identify the objects or events which match, the task of matching becomes much more difficult. Thus, in the automotive example given above, if order number 123456 calls for a high performance engine but the engine is

coded 12345; either because of a mistake or because of a systematic problem (e.g. the engine encoder can only specify 5 digits) it may or may not be possible to deterministically match the engine to an order. Assuming that it is desirable that orders be filled in the normal process where possible, but more important that no unordered cars be assembled; the need for a deterministic matching method can be seen.

**[0003]** An actual example of a process in which a deterministic matching method can be used to good effect is the mail tracking program recently introduced by the United States Postal Service (hereinafter sometimes "USPS") under the trademark CONFIRM®. The CONFIRM program tracks mailpieces through the delivery process and provides the mailer with information about the status of the mailpiece in the delivery process.

**[0004]** Figure 1 shows a simplified representation of the USPS mailpiece delivery process incorporating the CONFIRM mail tracking process. Mailers 10 deliver large mailings 12 to local postal facilities 14. Mailings 12 can comprise letters, flats, periodicals, or the like (sometimes hereinafter generally "letters") and are typically pre-sorted to various degrees by mailers 10 to obtain discounts offered by the USPS. (While the general principles of the CONFIRM process can be applied to any mailpiece or package, as currently implemented by the USPS it used for tracking large, typically pre-sorted, mailings and reply mail returned in pre-addressed and pre-coded business reply envelopes (sometimes herein "BRE's") sent out in such mailings.)

**[0005]** At facility 14 scanner 16 scans mailings 12 to establish the times at which mailings 12 entered the mail delivery process. Typically individual letters are not scanned at facility 14, though they can be if appropriate scanners are available, and accompanying documents(not shown) are scanned to establish the time and facility at which each of mailings 12 entered the delivery process and/or the times at which mailings 12 are sent on in the delivery process. This information is then sent to programmable data processing system (hereinafter sometimes server) 22 over one of communications links 20. (Links 20 connect various postal facilities to server 22 and can take any convenient form or forms; details of which form no part of the subject invention.) Mailings 12 and other mailpieces (not shown) are then grouped to form assemblies 28 which are then delivered to sectional postal center 30. At center 30

assemblies 28, together with other assemblies 28 from other facilities (not shown) are processed by conventional high speed scanner/sorters 32. As letters are sorted and otherwise processed at center 30 a POSTNET code (commonly known as a “zipcode” and hereinafter sometimes “pn”) and a code used in the CONFIRM process known by the USPS trademark PLANET Code® (hereinafter sometimes “pc”) are scanned from the letters. The pair <pn, pc> identify, though not necessarily uniquely identify, the letters within a set of letters, and events associated with those letters, as will be described further below. In accordance with USPS requirements the POSTNET Code and PLANET Code are printed as barcodes. As letters are scanned, typically for the first time at center 30, the values pn and pc are sent to sever 22 over one of links 20, where they are combined with information about mailings 12 from facility 14, as will be described further below. (Note that the PLANET Code for a letter will identify a corresponding mailing so that the time a letter entered the mail delivery process can be determined by sever 22 from information provided by facility 14.)

**[0006]** Sorted assemblies 36 are delivered to local postal facilities 38 for further sorting and delivery of letters /. Letters / can be scanned at facilities by scanners 40 and information including values for pn and pc sent to server 22 over one of links 20. BRE’s included in mailings 12 can be returned by the receiving customers and are scanned, sorted and delivered in substantially the same manner as described above, except that BRE’s are typically returned as individual letters and, of course need not follow the same path through the postal system as mailings 12.

**[0007]** As a letter moves through the delivery process and various steps or stages occur (hereinafter sometimes “events”) and these events are recorded as an n-tuple <d, f, t, pn, pc> where:

d is the time at which the event occurred;

f is the facility at which the event occurred;

t is the type of event;

pn is the POSTNET (possibly partial) scanned from the letter when the event occurs; and

pc is the PLANET Code scanned from the letter when the event occurs.

Server 22 is programmed to match these events with letters having corresponding values for the pair <pn, pc> and maintain database 24 of matched letters and events. Mailers 10 can then access database 24 through server 22 and network 42, or in any other convenient manner, to determine the status of letters in the mail delivery system. However the <pn, pc> data is often incomplete or inaccurate. Further, different letters may have the same values for <pn, pc>. Thus it is not always possible to uniquely match an event and a corresponding letter. Since mailers would rather have no tracking information than incorrect information, there is a need for a deterministic method and system which will match letters and events in the context of the USPS CONFIRM system. Current methods for matching letters and events attempt to match only those cases where the full (11 digit) POSTNET code and PLANET Code are scanned and allow matching of approximately 80 to 85% of events to letters; rejecting the remaining 15 to 20%.

**[0008]** Program code to control server 22 in accordance with the subject invention is provided on magnetic or optical disk 46-D and input through disk reader 46-R, or through any other suitable computer readable medium. The term "computer-readable medium" as used herein refers to any medium that participates in providing program code to a processor for execution. Such a medium may take many forms, including but not limited to, non-volatile media, volatile media, and transmission media. Non-volatile media includes, for example, optical or magnetic disks, or random access read only memory. Volatile media includes random access dynamic memory. Transmission media includes coaxial cables, copper wire and fiber optics. Transmission media can also take the form of acoustic or light waves, such as those generated during radio wave and infrared data communications. By "program code" herein is meant sequences of indicia recorded on, or signals transmitted by, computer readable media, which, when input by a processor cause a processor to carry out a corresponding sequence of operations, i.e. execute the program code.

**[0009]** In Figure 2 letter / includes a delivery address printed in block 50, a return address printed in block 52, a stamp or postal indicium indicating payment of postage 54, PLANET Code barcode 56, and POSTNET barcode 58. Letter / can be printed with additional markings which do not relate to the subject invention and can be arranged in

somewhat different formats without affect on the teaching of the present application. Barcode 58 typically represents 5, 9, or 11 digits which specify delivery information with increasing precision. The code used is typically determined by the information available to mailer 10 or by the extent which mailer 10 wishes to pre-sort mailings 12. PLANET Code 56 can take on one of two formats, format 56-D to track delivery of letters in mailings 12 and format 56-O to track BRE's returned by customers. Format 56-D specifies the service (e.g. first class letters, standard mail flats, etc.) subscriber (i.e. mailer) id, and mailing id. Typically there is one PLANET Code used in a mailing. Format 56-O specifies the service and a customer id. The CONFIRM system relies on the BRE's POSTNET code to identify the mailer.

**[0010]** While generally the POSTNET and PLANET Code, i.e. the pair<pn, pc>, uniquely specify a letter it is apparent that this need not be so. For example, if a mailer 10 chooses not to sort a mailing 12, letters in that mailing may only have a 5 digit POSTNET code with many letters having the same value for pn. Or scanners at various facilities may not have a capability to scan a full 11 digit POSTNET code so that a event record can not always be uniquely associated with a letter record. More rarely, even the full POSTNET code will not distinguish all letters, as when two letters in a mailing are sent to the same address. (Since the CONFIRM system is relatively new it is believed all PLANET Code scanners can scan the full PLANET Code.)

**[0011]** (The above simplified description of the mail delivery process and the CONFIRM system is given to provide background and those skilled in the art will recognize that substantial differences from actual mail delivery can exist. Such differences however are not believed to affect the present teaching. Further details of the CONFIRM system are found in: The CONFIRM® System Customer Service Guide, United States Postal Service, publication no. 197, September 2002.)

**[0012]** Thus it is an object of the subject invention to provide a method and system for deterministic matching of objects and events, such as letters and events occurring as the letters are delivered, even though the letters and events are not necessarily uniquely identified.

### **Brief Summary of the Invention**

**[0013]** The above object is achieved and the disadvantages of the prior art are overcome in accordance with the subject invention by a method, and a system for carrying out that method. The method deterministically matches first elements of a first set of objects or events with second elements of a second set of objects or events, matching first and second elements each being associated with common values of an identification code  $pn$  having  $|pn|$  characters, and where the identification code can be insufficient to uniquely identify the first elements, and portions of the identification code values associated with the first and second elements can be unknown; and includes the steps of: a) generating a mapping  $\theta$  for the first set such that, for each element  $i_i$  of the first set  $\theta(i_i)$  equals  $\langle k_i, ppn_i \rangle$ , where  $pn_i$  is at least a portion of the identification code value associated with the element  $i_i$  and  $ppn_i$  is defined as the first  $k_i$  characters of  $pn_i$ , and  $k_i$  is selected to be the minimum number of characters required to uniquely identify  $i_i$  in the first set, whereby values for  $k_i$  greater than  $|pn|$  imply that the element  $i_i$  is not uniquely identified by the portion  $ppn_i$ ; b) determining  $pn_j$  for an element  $e_j$  in the second set, where  $pn_j$  is at least a portion of the identification code value associated with the element  $e_j$ ; and c) matching the element  $e_j$  and the element  $i_i$  only if the first  $k_i$  characters of  $pn_j$  equal  $ppn_i$  and not matching the element  $e_j$  and the element  $i_i$  if the element  $i_i$  is not uniquely identified in the first set by the portion  $ppn_i$ . The system can be controlled in accordance with program code on a computer readable medium.

**[0014]** In accordance with one aspect of the subject invention the first elements are letters and the second elements are events which occur during processing of the letters.

**[0015]** In accordance with another aspect of the subject invention the method includes a step of performing an additional consistency test and matching said letter  $i_i$  and event  $e_j$  only if said consistency test confirms such match.

**[0016]** In accordance with still another aspect of the subject invention the method includes: a) generating a minimal  $k$ -unique mapping for said first set such that, for each element  $i_i$  of said first set such that  $i_i$  maps to a pair  $\langle k_i, ppn_i \rangle$ , where  $pn_i$  is at least a portion of said identification code value  $pn$  associated with said first elements and  $ppn_i$  is defined as the first  $k_i$  characters of  $pn_i$ , b) determining  $pn_j$  for an element  $e_j$  in said

second set, where  $pn_j$  is at least a portion of said identification code value associated with said element  $e_j$ ; and c) matching said element  $e_j$  and said element  $l_i$  only if the first  $k_i$  characters of  $pn_j$  equal  $ppn_i$  and not matching said element  $e_j$  and said element  $l_i$  if said element  $l_i$  is not uniquely identified in said first set.

**[0017]** Other objects and advantages of the subject invention will be apparent to those skilled in the art from consideration of the detailed description set forth below and the attached drawings.

### **Brief Description of the Drawings**

**[0018]** The subject invention is illustrated by way of example, and not by way of limitation, in the figures of the accompanying drawings and in which like reference numerals refer to similar elements and in which:

**[0019]** Figure 1 shows a schematic block diagram of a simplified representation of a postal delivery system.

**[0020]** Figure 2 shows a schematic representation of a plan view of a letter mailpiece.

**[0021]** Figure 3 shows a schematic representation of a database of letters and matched events which occur during the delivery of the letters.

**[0022]** Figure 4 shows a flow diagram of the operation of a server in updating or generating a minimal  $k$ -uniquely identified mapping on a set of letters  $L$  in accordance with one embodiment of the subject invention.

**[0023]** Figure 5 shows a flow diagram of the operation of a server in matching events  $e$  in a set of events  $E$  to letters  $l$  in a set of letters  $L$  in accordance with one embodiment of the subject invention.

### Detailed Description of Preferred Embodiments of the Invention

**[0024]** Figure 3 shows database 24 of letters  $l$  and events  $e$ . File 60 includes records 62-1 through 62-M,  $M = |L|$ , and corresponds to a set  $L$  of letters  $l$ . Records 62 include fields  $l$  which contain n-tuples  $\langle id, pn, pc, s, ed \rangle$  where:

$id$  is a unique identifier for the n-tuple generated by sever 22,

$pn$  is a POSTNET value,

$pc$  is PLANET Code value,

$s$  is the time corresponding physical letters entered the mail process (i.e. the time that the corresponding mailing 12 was received at facility 14), and

$ed$  is the estimated latest time of delivery of corresponding physical letters; and fields 64 contain the pairs  $\langle k, ppn \rangle = \langle k, pre(pn, k) \rangle$  (where  $pre(pn, k)$  is defined as the first  $k$  digits of  $pn$ ) which map to the pair  $\langle pn, pc \rangle$  such that letter  $l$  is minimally  $k$ -uniquely identified in  $L$  by  $\langle pn, pc \rangle$ , as will be describes further below.

**[0025]** Latest time of delivery  $ed$  is determined by the USPS in a known manner in accordance with factors such as the type of mail, degree of pre-sorting, etc., and is typically 7 to 14 days. The pair  $\langle s, ed \rangle$  define a life-span of a letter and if a letter is not delivered within its life-span it is unlikely that it will be delivered at all.

**[0026]** File 66 includes records 68-1 through 68-M, and corresponds to events  $e$  in set  $E$  which are matched to letters  $l$ , as will be described further below. Records 68 include fields 70 which contain a value  $id$  identifying a letter  $l$  and pluralities of fields 72, each field in pluralities 72 containing an event  $e_x$  to indicate that events  $e$  recorded in one of records 68 are matched to the letter  $l$  identified in that record. In Figure 3 event  $e_j$  has been expanded to show n-tuple  $\langle d_j, f_j, t_j, pn_j, pc_j \rangle$ , as described above.

**[0027]** As discussed above, in general it is not always possible to match letters  $l$  and events  $e$ . Letter  $l$  may not be uniquely identified by pair  $\langle pn, pc \rangle$ , or only a portion of  $pn$  may be scanned during the event so that insufficient information is available to uniquely match the event to a letter. Such unmatched events can be handled in any convenient manner, details of which form no part of the present invention.



**[0028]** Those skilled in the art will recognize that database 24 can be organized in other ways, and details of the organization of database 24 form no part of the subject invention.

**[0029]** Figure 4 shows a flow diagram of the operation of server 22 in updating or generating a set  $L$  which is minimally  $k$ -uniquely mapped by a mapping  $\theta$  such that  $\theta(l_i) = \langle k_i, \text{ppn}_i \rangle$ . Set  $L$  is defined as  $k$ -uniquely mapped if for each letter  $l_i$  in  $L$ :  $l_i \rightarrow \langle k_i, \text{ppn}_i \rangle$ ,  $k_i$  an integer such that  $\min|\text{pn}| \leq k_i \leq \max|\text{pn}|$  (where preferably  $\min|\text{pn}| = 5$ ,  $\max|\text{pn}| = 11$ ), if and only if for all letters  $l_i, l_j$  in  $L$ ,  $i \neq j$ ,  $\langle \text{pc}_i, \text{pre}(\text{pn}_i, k_i) \rangle \neq \langle \text{pc}_j, \text{pre}(\text{pn}_j, k_j) \rangle$ . Set  $L$  is defined as minimally  $k$ -uniquely mapped if  $L$  is  $k$ -uniquely mapped, and for all letters  $l_i, l_j$  in  $L$ ,  $k_i > |\text{pn}|$ ,  $\langle \text{pc}_i, \text{pre}(\text{pn}_i, k_i-1) \rangle = \langle \text{pc}_j, \text{pre}(\text{pn}_j, k_j-1) \rangle$ . That is: if set  $L$  is minimally  $k$ -uniquely mapped then for each letter  $l_i$   $k_i$ , if it exists, specifies the minimum number, greater than  $\min|\text{pn}|$ , of initial digits of  $\text{pn}_i$  which are sufficient to distinguish  $\langle \text{pc}_i, \text{pn}_i \rangle$  from all other  $\langle \text{pc}_j, \text{pn}_j \rangle$ , and thus distinguish  $l_i$  from all other letters  $l_j$  in  $L$ .

**[0030]** (It should be noted that there is a one-to-one correspondence between  $n$ -tuples contained in fields  $/$  and corresponding physical letters, and a similar correspondence between  $n$ -tuples  $\langle d, f, t, \text{pn}, \text{pc} \rangle$  and events, and that what are actually matched in accordance with the subject invention are corresponding records in database 24. Accordingly, except where the difference may be both significant and unclear, the terms "letters" and "events" are used herein to refer to both physical letters and events and the corresponding  $n$ -tuples,  $\langle \text{id}, \text{pn}, \text{pc}, s, \text{ed} \rangle$  and  $\langle d, f, t, \text{pn}, \text{pc} \rangle$ , respectively.)

**[0031]** Initially at step 80 in Figure 4 letter set  $L$  includes letters  $l_j$ ,  $j = 1, 2, \dots, |L|$  or may be empty. Letters  $l_j \rightarrow \langle k_j, \text{ppn}_j \rangle$ ; where  $k_j$  is a positive integer representing a previous minimal  $k$ -unique mapping of  $|L|$ . The index  $k$  is initially set equal to  $\min|\text{pn}|$ , which is preferably equal to 5. (Set  $L$  is assumed to have been previously sorted so that for all letters  $l_i, l_j$  in  $L$ , it can be assumed, without loss of generality, that  $\text{pc}_i = \text{pc}_j$ .)

**[0032]** After starting, at step 82 server 22 sets  $m=1$  and at step 84 inputs a new letter  $l_m = \langle \text{id}_m, \text{pn}_m, \text{pc}_m, s_m, \text{ed}_m \rangle$ . For ease of description letter  $l_m$  is assumed to be input as it is scanned for the first time, as would be the case, for example, with a BRE. For letters in mailings it can be that the life span,  $\langle s_m, e_m \rangle$ , is determined when the mailing is delivered to facility 14, as described above, and server 22 will match letter  $l_j$  to

the appropriate life span in accordance with  $pc_m$  when letter  $l_j$  is first scanned separately. This operation would be well within the abilities of one skilled in the art and need not be described further here for an understanding of the subject invention. In either case, at step 84 an arbitrary unique identifier  $id_m$  is added to field  $l_m$  in record 62-m (shown in Figure 3). This identifier is used to identify record 62-m in database 24 and is typically not included on physical letters  $l$ .

**[0033]** Then at step 86, server 22 tests to determine if  $m = 1$ , and, if so, at step 90 maps  $l_m = l_1$  to  $\langle 5, ppn_m \rangle$ , since at this point only  $l_1$  is in  $L$ . Then at step 92 server 22 sets  $m = m+1$ , and at step 94 determines if there are any more letters to be input. If not the program ends and, otherwise returns to step 84.

**[0034]** Thereafter, for each letter  $l_m$ , at step 98 server 22 sets  $j = 1$  and at step 100 tests to determine if  $j = m$  and if not at step 102 tests if  $pre(pn_j, k) = pre(pn_m, k)$ . That is, if the first  $k$  characters (initially 5 characters) of  $pn_j$  equal the first  $k$  characters of  $pn_m$ . If so, at step 106 the equals flag, EQF, is set equal to 1, and at step 108 server 22 tests to determine if  $k_j < k+1$ , and if so, at step 110 sets  $k_j = k+1$ ; since step 102 has shown that at least  $k+1$  characters are necessary to distinguish  $pn_j$  from  $pn_m$ . Then, or if  $k_j$  is less than  $k+1$ , or if at step 102  $pre(pn_j, k) \neq pre(pn_m, k)$ , at step 112 server 22 sets  $j = j+1$  and goes to step 100.

**[0035]** If at step 102  $|pn_j|$  or  $|pn_m|$  are less than  $k$ ,  $pn_j$  or  $pn_m$  are filled out with unknown characters "?" which are considered to match any character. For example, for  $pn = 1,2,3,4,5$   $pre(pn, 7) \equiv 1,2,3,4,5,?,?$  and is considered to equal  $1,2,3,4,5,x,y$ ; where "x" and "y" can be any character, including "?".

**[0036]** If at step 100  $j = m$ , server 22 goes to step 114 to determine if EQF is set. Since  $l_m$  has been tested at step 102 against  $l_j$  for all  $j$ ,  $j < m$ ; and if EQF is not set then the current value of  $k$  is sufficient to uniquely distinguish letter  $l_m$  from all letters  $l_j$  in  $L$ , then if EQF is not set at step 120  $k_m$  is set equal to the current value for  $k$  and  $l_m \rightarrow \langle k_m, ppn_m \rangle$ , i.e.  $\langle k, ppn_m \rangle$  is stored in field 64-m (shown in Figure 3) and  $L$ , as currently constituted, is minimally  $k$ -uniquely mapped. Then at step 122 EQF is reset and  $k$  is set equal to 5, and server 22 then goes to steps 92 and 94 and either exits or inputs a next letter, as described above.

**[0037]** If EQF is set at step 114, then at step 124 server 22 determines if  $k = \max|pn|$  (preferably  $\max|pn| = 11$ ) and, if not, at step 126 sets  $k = k+1$  and resets EQF and goes to step 98 to repeat the cycle described above for the next value for  $k$ . If  $k = \max|pn|$ , then at step 130  $k_m$  is set equal to  $\max|pn|+1$  and  $l_m \rightarrow \langle k_m, X \rangle$ , i.e.  $\langle \max|pn|+1, X \rangle$ , where  $X$  is arbitrary, is stored in field 64- $m$ ; indicating that for at least one  $j$ ,  $j \neq m$ ,  $pn_j = pn_m$  (and thus that  $l_j$  and  $l_m$  are not uniquely identified) and then goes to steps 122, 92, and 94, as described above.

**[0038]** (A file of previously scanned, unmapped letters  $l$  can be mapped by modifying the operation of server 22 as shown in Figure 4 by:

setting all  $k_j \leq \min|pn|$   
 setting  $m = 1$  at step 82, and  
 deleting step 84.)

**[0039]** Figure 5 shows a flow diagram of the operation of server 22 in matching events  $e$  in set **E** to letters  $l$  in set **L**. At step 150 server 22 waits for the occurrence of an event in the delivery process. When an event  $e_j$  occurs during delivery of a letter  $l$  server 22 inputs event  $e_j = \langle d_j, f_j, t_j, pn_j, pc_j \rangle$  and sets  $i = 1$  at step 152. At step 154 server 22 gets  $l_i$  and  $\Theta(l_i)$ ; that is, it obtains fields  $l_i$  and 64- $i$  in record 62- $i$  in file 60 (shown in Figure 3).

**[0040]** Tests at steps 156 through 164 are then applied to determine if event  $e_j$  is uniquely matched to letter  $l_i$ . At step 156 it is determined if  $pc_j$  equals  $pc_i$ . If so, at step 160 it is determined if  $k_i$  is less than or equal to  $|pn_j|$ ; that is if there are sufficient characters in  $pn_j$ , as scanned, to uniquely match  $e_j$  and  $l_i$ . Note that this test prevents a match if  $k_i$  is greater than  $\max|pn|$ ; i.e.  $l_i$  is not uniquely identified in **L**. If so, then at step 162 it is determined if  $pre(pn_j, k_i)$  is equal to  $ppn_i$ ; that is if the first  $k_i$  characters of  $pn_j$  and  $pn_i$  are equal; which serves to uniquely establish a match. Then, as a consistency check, at step 164 it is determined if  $s_i < d_j < ed_i$ ; that is if event  $e_j$  has occurred during the life period of letter  $l_i$ . If all the above tests are passed then at step 168 event  $e_j$  is uniquely matched to letter  $l_i$ ; preferably by storing  $\langle d_j, f_j, t_j, pn_j, pc_j \rangle$  in field  $e_j$  in record 68- $i$  so that it is associated with identification  $id_i$  in field 70- $i$  (shown Figure 3).

**[0041]** Approximately 2 to 8% of letters which should have a full POSTNET code are initially produced or described by mailers with less than the full 11 digits. In at least

some cases the USPS has the capability to correct this and provide the full POSTNET code, with the result that  $|pn_j|$  will be greater than  $|pn_i|$ . (The manner in which this correction is made forms no part of the subject invention and need not be described further here.) Accordingly, in the preferred embodiment shown, at step 170 server 22 determines if  $|pn_j| > |pn_i|$ , and, if so, at step 172 sets  $pn_i = pn_j$  so that the full, or extended, value can be used for matching later, downstream events.

**[0042]** Then at step 186 server 22 determines if any more events are pending, and if so returns to step 152 to process the next event, and otherwise exits. If any of the above described tests are not passed then at step 176 server 22 determines if  $i$  equals  $M$ ; that is if all letters  $l$  have been tested. If so, at step 178 a no match is recorded for event  $e_j$  in any convenient manner and server 22 goes to step 170 and continues as described above. Otherwise  $i$  is set equal to  $i+1$  at step 180 and server 22 returns to step 154.

**[0043]** In other embodiments of the subject invention other consistency tests can be used in addition to or in place of the test of step 164. For example the type  $t_j$  and location  $d_j$  can be tested for consistency; e.g. automatic sortation at a facility lacking such equipment.

**[0044]** Those skilled in the art will recognize that where event  $e_j$  is the entry of a letter  $l_i$  into the delivery process then it is only necessary to start a new record 68- $i$  in file 66 including entry event  $e_j$ .

**[0045]** Returning to Figure 1, when one of mailers 10 wishes to determine the status of a letter  $l$  in a mailing, or if a BRE in a mailing has been returned, an inquiry can be submitted to server 22 through network 42. Server 22 can then access database 24 to identify relevant events which have occurred; and from the above description it will be apparent to those skilled in the art that events will be identified if and only if they are deterministically matched to a letter.

**[0046]** The embodiments described above and illustrated in the attached drawings have been given by way of example and illustration only. From the teachings of the present application those skilled in the art will readily recognize numerous other embodiments in accordance with the subject invention. Particularly it will be apparent to those skilled in the art that the sequences of steps and data structures described above

can be altered in numerous logically equivalent ways without departing from the scope of the subject invention. Accordingly, limitations on the subject invention are to be found only in the claims set forth below.